Wireless Protocol Security: to Simulate or not Simulate?*

Todd R. Andel and Alec Yasinsac**
Department of Computer Science
Florida State University
Tallahassee, FL 32306-4530
{andel | yasinsac}@cs.fsu.edu

Keywords: Ad hoc routing, proving security, formal methods

Abstract

Secure routing protocols for mobile ad hoc network are vital to proper wireless network operations. Unfortunately, the security properties of these protocols is often unknown and difficult to analyze. There are various techniques available to evaluate security properties, to include: coordinated discussion, simulation, experimental observation, analytical proofs, and formal methods. These approaches can only be used to provide operational insight and to analyze routing security weaknesses or vulnerabilities. None of the approaches can prove that a protocol is unconditionally secure, since proving security is an intractable problem. In this paper we discuss the capabilities and limitations of these general approaches for use in evaluating security properties of wireless routing protocols.

INTRODUCTION

Mobile ad hoc networks (MANETs) face significant security challenges due to the inability to confine a wireless radio signal. Route security is vital to the proper operation and reliability of such protocols. If a malicious host can inject itself into the routing path, proper operation of the routing protocol is dependent on the attacker's intentions.

Unfortunately, there seems to be a wide gap between providing security and the development of MANET routing protocols. Security is either ignored or security mechanisms are added as vulnerabilities or attacks are discovered. While this itself is not an ideal practice, it further complicates determining the security of such protocols. Following a formalized software engineering approach allows researchers to use various techniques to verify if a protocol's security goals stated in the protocol requirements have been met. Without specific security goals the question often becomes, "Is this protocol secure?" Unfortunately, proving complete security of a routing protocol is an intractable problem; there is no way to analyze a protocol to determine if it is vulnerability free against any yet undiscovered attack.

This paper examines the problems that exist in proving routing protocol security. This research proposes a composition approach to analyze the security or identify vulnerabilities in MANET routing protocols. We discuss five security evaluation approaches: coordinated discussion (walkthrough), simulation, experimental observation (testing), analytical proof, and formal methods. We are interested in what each of these can tell us about security.

ANALYZING SECURITY PROPERTIES

Perfect security is impossible to achieve, in fact, it is not clear that we can describe what that term means. So, it may not be surprising that neither are there effective mechanisms that measure system security. Security researchers seek solutions, similar to those from other disciplines, that are reasonably easy to compute and are well-validated through years of observation and analysis. For example, reliability research recognizes *system availability* (time in service/total time) and *mean time to failure* (total time/# failures) as widely applicable metrics that tend to reflect reality.

There have been attempts to produce a comprehensive security metric based on practice, probability, and simulation [1, 2], but no single metric or group of metrics can comprehensively capture information security properties. Rather, we discuss security properties in terms of the threat picture, essentially describing how individual security techniques address particular attacks.

Most security research characterizes their effectiveness in isolation and their mechanisms are measured by unrelated approaches. For example, encryption algorithms may be provably strong against polynomial adversaries who employ

* This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under grants numbered DAAD19-02-1-0235 and W91NF-04-1-0415.

** The views expressed in this article are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

chosen plaintext attacks, a cryptographically specific and meaningful definition. Conversely, formal methods are the most commonly used approach to verify security protocols. Though there have been recent attempts to bridge this gap [3], there is no accepted method to combine these mechanisms to get a comprehensive security picture.

We contend that the five common security verification mechanisms above each have different, often complimentary properties. We suggest how to leverage this complementation to achieve more comprehensive security evaluation and to facilitate understanding regarding how each approach fits in to a coordinated security verification evaluation.

We begin by structuring our argument around attack verification categories. There has been a litany of work to address attack types, for example in replay attacks [4] and denial of service attacks [5]. We, however, are interested in verification approaches relative to pertinent attacks. To illustrate this concept, we first address the oldest verification approach: coordinated discussion. In its more primitive form, this technique reduces to human analysis, possibly consisting of a single analyst sitting down with a protocol specification and a pencil. Modern verification research recognizes the importance of this technique, and has evolved it into the oft-dreaded *structured walkthrough.* Because of its simplicity, we use the walkthrough approach to illustrate our attack verification categories.

In our first verification category, we ask whether an approach can verify unknown attack prevention/detection, or if it is limited to only known attacks. This is one of the walkthrough approach's strengths. Because human thought is creative, walkthroughs can identify previously unknown vulnerability. This may be facilitated by spending part of the walkthrough session examining possible attacks and allowing minds to wander. When members identify unknown (or known) attacks, they may evaluate the attack's effectiveness and potential impacts on the spot. Walkthroughs allow breadth of thought and interaction that, while it may lead to time wasted discussing remotely related items, it also often leads to otherwise difficult or impossible to reach conclusions.

We also consider verification approaches in terms of whether they provide property guarantees, or whether they only provide security property evidence. Walkthroughs can result in wide and deep analysis, but cannot provide complete verification for even simple properties. Information system developers (maybe the most common walkthrough users) well-recognize this limitation. However, they routinely conduct walkthroughs because walkthroughs identify problems that may be difficult and expensive to find and fix later. This applies to security analysis as well.

Most modern development paradigms recognize that verification begins with the initial system conception and continues until a system's death. However, the most intense verification activity occurs between system specification and implementation. The third verification category is whether an approach can be used before an implementation is available. A second strength of walkthroughs is that they are potentially effective in any development phase and are routinely used to verify security protocol conceptualization, specification, and design[1].

Finally, we consider the rigor that the verification approach reflects, essentially if the approach is dependent on the context. Walkthrough results are often presented as prose descriptions and summaries, possibly reflective of human genius, but always subject to human misunderstanding. Their meaning must always be evaluated within their intended context, so walkthrough results must be considered to be context dependent.

Table 1 summarizes walkthrough verification approach categories. The rest of this paper will fill in the table for the other protocol verification approaches. We focus particular effort, time, and discussion on simulation because our experience reflects broad misunderstanding regarding simulation's capability to capture or represent security properties and notions (including some of our own).

Table 1. Walkthrough Characteristics

Verification Approach	Unknown Attacks	Property Guarantees	Design Phase	Context Dependent
Walkthrough	yes	no	yes	yes

SIMULATION

MANET routing protocol development typically relies on simulations modeled in a commercial, open source, or an independently developed simulation package. Simulation is vital, since much of this research area is in the early stages and has not yet been physically implemented. In lieu of testing, simulation provides a projection or approximation of the protocol operation. Simulation-based study projecting average case performance based on statistical analysis of independent runs has become the de facto standard for network research and is not only an accepted validation approach, but is a prerequisite for research result publication.

Still, a wide range of factors can affect simulation results, and an improperly validated simulation can be misleading. Identical studies performed within different simulation packages have been shown to produce inconsistent results [6]. Additionally, physical layer modeling abstraction of wireless radio waves [7], improper

[1] Conversely, execution testing occurs when executable artifacts are available. More on this later.

statistical procedures [13], and incomplete documentation have significant impacts on simulation studies. Ultimately, each simulation must be validated for its intended environment. There are many approaches to validating simulations. One way is to corroborate simulation models against actual implementations, i.e. to simulate concept specifications and then view implementations created from the same specification and compare the results. If scientific rigor is applied, the simulation approach validity may be illuminated.

A second way to validate a simulation approach is to compare simulation results to a different verification approach. Corroborating simulation results through analytical proofs can add substantial credibility to the verification result over simulation alone.

Clearly simulation is a powerful research tool with two major advantages:

(1) Simulation allows a researcher to examine an idea's properties without implementing (or constructing) that idea. Many different samplings can be evaluated and configurations can be adjusted before and during simulations to facilitate analysis and to optimize the resulting product. While simulation is not free, in most cases, it is much less expensive to simulate first then develop than it is to develop and test a concept without simulation. If this is not the case, simulation should not be undertaken. Fortunately, evolving industrial and commercial simulation packages and frameworks often ensure easy and inexpensive simulation capabilities.

(2) Simulation allows the investigator to isolate sections, components, subsystems, etc. during evaluation By holding control items constant and varying specific, targeted values, the investigator may more accurately examine specific, systemic cause and effect relationships.

Unfortunately, these two advantages also lead directly to the two most common simulation pitfalls. First, because simulation is relatively inexpensive and easy (particularly after the initial learning curve is overcome) investigators are prone to misuse simulation in situations where it is not scientifically suited. This is a major point of our paper and we discuss this issue more later.

Secondly, in modern information systems, there are few components that operate in isolation. Isolated simulation parameters may not accurately reflect how the system will operate when components are interacting. Of course, simulation is only valuable if it accurately projects how constructed systems will operate in the real world (i.e. after they are constructed). Simulations that reflect isolated properties can help an analyst to optimize those properties, but the analyst must consider how the isolated adjustments will impact overall system operation.

In one sense, simulation is nothing more than testing of an approximation of the proposed system. The only properties that can be tested during simulations are those that are approximated in the simulation. Accordingly, simulations are limited to testing attacks that are known a priori. Additionally, since simulations do not reflect the precise operating environment, they are dependent upon their execution context.

A final strength of simulation, particularly computer simulation, is that repetitive testing with large data volumes is usually possible and inexpensive. It is this characteristic that encourages companies to maintain computer simulations of existing systems, i.e. to allow maintenance engineers to project change impacts without implementing the potential changes. This characteristic is also valuable during initial development, particularly for generating performance estimates that lend themselves to probabilistic analysis. Because simulations are statistically driven, performance is an effective simulation target.

Simulation and Security

Though simulation has the potential to accurately estimate or project network performance, it is not as clear how simulation can help to estimate network protocol security. One security-related area where simulation has shown promise is Intrusion Detection. For example, [8] contains a report of an experiment simulating one hundred and twenty hosts operating concurrently, using only five real hosts (each real host simulated twenty four experimental hosts). In this case, the production software is in place, but the expected (or worst case) number of hosts is not. Thus, the author uses simulation to give a system assessment, although non-trivial artificiality remains.

As far as network simulation goes, this environment is relatively similar to the projected real operating environment. Conversely, mathematical, or computer simulations such as NS-2, have only theoretical, sometimes artificial, correlations with real environments. As a rule, the less artificiality a simulation contains, the more credible its results are.

Another example of intrusion detection simulation comes from Defense Advance Research Project Agency (DARPA) intrusion detection systems research [9, 10]. In this work, MIT Lincoln Laboratory tested various intrusion detection systems against simulated network traffic, consisting of normal background traffic and attack traffic. The systems were rated according to the percentage of attacks detected vs. the false alarms per day.

These evaluations lean more toward testing than simulation, since they involve mostly functional systems. The simulation exists in traffic formation. If real world traffic was fed into the target systems, we would consider this "beta testing". Conversely, if purely synthetic traffic was used, we would consider this a testing process. However, actual traffic was simulated real world traffic with

inserted attack streams and obfuscated addresses. The traffic was synthetically generated to statistically simulate real network traffic. The attack traffic consisted of various types of attacks that were both well known and some were new at the time of testing.

The only properties we can determine with this type of security testing is how these systems perform against these specific simulated attacks in this specific environment (i.e. the simulated background traffic). These results cannot determine the effectiveness of any of these systems against any other attacks or different network profiles other than those that were tested. While one system may work well under a given set of conditions (background traffic and attack type), it may operate completely differently under different network conditions and different attacks than which it was tested against.

Security in MANET Simulations

To date, we are yet to see a simulation that reflects MANET protocol security properties. Further, we contend that exercise of a MANET routing protocol against a simulated attack will only provide information on how well the protocol defended against the simulated attacks under the simulated conditions. However, these simulations can tell us very little protocol security properties under actual operating conditions.

Awerbuch et al. propose an example of a MANET simulation to determine the performance of two wireless routing protocols under various Byzantine attacks [11]. These include black hole, flood rushing, and wormhole attacks in wireless networks in which all nodes are authenticated. The threat of Byzantine attacks, one in which a node with appropriate authentication and keying material is captured or obtained by an adversary, is a significant threat in wireless ad hoc networks.

A Byzantine attacker can participate in the routing protocol virtually undetected since it is an authorized node. Once a malicious node is chosen as part of a data route, it can drop packets during data transmission. The authors [11] present the On-Demand Secure Byzantine Routing (ODSBR) protocol, which was developed to defend against such attacks. Simulation is used to show how ODSBR and Ad hoc On-demand Distance Vector (AODV) perform against various Byzantine attacks, while differing the number of attackers. The simulation shows that ODSBR is less susceptible to packet loss than AODV under these attacks. It also depicts that intelligent node placement (i.e. a malicious node in the center of the network) increases the probability that an attack will succeed.

According to the simulation results, one may view ODSBR as more secure that AODV. This is not necessarily a true statement, the packet delivery rate of ODSBR was more resilient to attacks than AODV in the simulated scenario only. These protocols may react differently under real-world implementations or under different types of attacks. Hence, simulation can only provide results for the attacks and the environment simulated for that scenario.

Moreover, the simulation essence in these cases again reflects the security property performance impacts. Security verification should reveal the mechanism's security properties relative to the target threat picture.

In addition to performance analysis, simulation can assist in attack visualization. For instance, simulating a worm attack may provide insight into how a given worm may infiltrate and spread across a network. Unfortunately, the predominate network research simulation tool, NS-2, does not provide adequate facilities for simulating MANETs to this end. The NS-2 visualization tool, Network Animator (NAM), is not intended for wireless use [12]. It shows wireless transmission ranges as footprint circles, but does not actually show how a packet propagates across a wireless network. The Colorado School of the Mines developed an independent visualization tool called the interactive NS-2 protocol and environment confirmation tool (iNSpect) [13]. This tool provides animations to allow node mobility tracking and visualizes both successful and unsuccessful packet transmissions between wireless nodes within NS-2 simulations.

As we have noted, correctly performed simulation can project MANET routing protocol average-case performance. In many cases, average-case performance analysis is acceptable, since outliers or statistical anomalies do not noticeably affect general protocol performance. Conversely, security must focus on "worst case" issues. That is, the attacker only has to find a single entry point to completely compromise a system. Simulating security vulnerabilities, or malicious attacks, can only be as precise as the data programmed into the simulation. While this method merits consideration in understanding and studying known attacks, it does not provide insight into how well a protocol deals with classes of attacks or unknown attacks.

Additionally, like testing, simulation cannot ensure that no vulnerability exists in analyzed protocols. Instead, we can use simulation to discover/detect the specific attacks programmed into the simulation, which can be used as evidence of modeled security properties.

OTHER WAYS TO ANALYZE SECURITY

Experimental observation, analytical proofs, and formal methods provide additional means to study MANET routing protocol security. As with simulation, none of these techniques can prove complete security, but each has characteristics that can improve security property understanding.

Experimental Observation

Protocol implementation and testing provides valuable data to determine protocol security. As protocols are developed, they can be tested against known attacks and active vulnerability assessments may discover security flaws. A more thorough approach of open source implementations allows other researchers to independently test protocols in operation and perform increased vulnerability analysis. Relating to our verification categories, formal testing rarely determines unknown attacks, since test-cases are designed with current attack knowledge. Additionally, testing can show evidence of security properties, but cannot provide guarantees on securities. Testing cannot begin until the protocols have been implemented and are then dependent on the environment in which they were tested.

Protocol realization and formal testing naturally leads to beta-testing by releasing to willing participants. Beta-testing does not follow a formalized test process with predetermined test cases; however, it allows for further assessment, increased protocol analysis, and a greater chance to encounter vulnerabilities (which may or may not be detected) that occur during normal protocol operation. The beta-test advantage is that unknown attacks may be detected before they product is widely distributed.

There is little information in the literature regarding implemented and tested MANET protocols, though MANETs are emerging in industry. This is due to the fact that most of the research in this area is still in early concept development. This however is not a "free-ticket" for simulation based research. MANET simulations should continue, but be balanced and validated against real-world implementations.

Analytical Proofs

We can also use mathematical proofs to verify the security properties of MANET routing protocols. Deductive proofs uses properties, theorems, and lemmas to prove or disprove security properties. Burmester et al. use this technique to confirm packet deliverability for various gossip protocols [14]. Essentially, they prove probabilistic broadcast delivery properties in the face of Byzantine threats. The authors use formal proofs to provide upper bounds on the propagation failure of these gossip protocols; however, limitations still exist. This technique can result in imprecision, since the proofs provide for upper bounds analysis and generally are used to compare orders of magnitude.

Also, as free-form arguments, proof techniques are dependent on assumptions and are prone to influence by unstated assumptions according to the mathematical abilities of the researcher, the proof complexity, and other environmental issues. Proofs must be tailored to an individual problem. For instance, two different researchers may attempt to prove a protocol property two completely different ways. The outcome can provide a reader with multiple or ambiguous interpretations of the problem.

One strong advantage of proof systems is that they can prevent unknown attacks by mathematically guarantying proven protocol security properties or goals. Proofs can be done at any time in the design phase or in operation if new attacks or vulnerabilities are discovered.

Formal Methods

Formal methods attempt to remove ambiguous proof results by eliminating a researcher's individual approach to deductive proofs. Formal methods implement a rigorous, standardized process to provide for unambiguous results. That is, two researchers following the same formal method should come to the same results and conclusions. The inclusion of formal methods analysis on MANET security properties in published research allow all readers to independently verify the results by following the rules of the given formal method.

We discuss the formal method BAN logic, which was introduced by Burrows, Abadi, and Needham in 1989 [15]. BAN was developed to evaluate trust relationships within authentication protocols. Evaluating a protocol in BAN follows four basic steps:

1. State the protocol goals
2. Convert the protocol to an "idealized" (BAN) form
3. State the assumptions
4. Apply BAN logic to the protocol messages and assumptions to derive the goals

Even though BAN logic follows a formalized process, errors can still be injected into the analysis. The technique of idealizing a protocol to the formal language is open to ambiguous transformations. Additionally, different assumptions may also alter the outcome.

Papadimitratos and Haas use BAN logic to analyze the security properties of their Secure Routing Protocol (SRP) [16]. SRP is an extension of the Dynamic Source Routing (DSR) protocol by attempting to guarantee that a malicious node is not part of a valid route between two end-points. These authors conclude that SRP is resilient to all attacks that do not depend on multiple cooperating malicious nodes. Disturbingly, this is not a true conclusion, Marshall et al. [17] show that a single malicious node can inject itself into the route path undetected to the to the BAN analysis performed by Papadimitratos and Haas.

This is not a failure of BAN logic, but shows how BAN logic was used to incorrectly analyze security properties of a MANET routing protocol. BAN truly was developed to analyze trust between authorized parties and not to analyze security properties in the face of malicious hosts [15]. Unfortunately, confusion still exists between the difference

of "trust" vs. security. We view BAN as a logic to show protocol correctness, which may alleviate this confusion.

This shows that formal methods can identify unknown attacks, while at the same time may not identify all possible unknown attacks. State protocol properties can be proven with correct formal methods approaches. As with analytical proofs, formal methods can be performed at any stage of protocol development or operation. BAN logic is dependent on the idealization step and assumptions, therefore, we consider this a semi-formal method. True formal methods, such as strand spaces [18] use unambiguous specification languages (no conversions required) and are not dependent an any surrounding context.

Model Checkers

Model checkers provide an automated extension to the formal methods approach. System properties are verified by an exhaustive search of all possible system states. Various model checkers (i.e. NRL protocol Analyzer [19], CPAL-ES [20]) have been used to show correctness of secure protocols. However, model checkers cannot prove absolute security due to the limitations in state-space explosion. Holzmann developed a model checker called SPIN that uses partial order reduction to reduce the number of possible states[21]. Further research is needed to determine if this technique could be adapted to analyze MANET routing protocols and the benefit, if any, this work would provide.

CONCLUSIONS

Can the security of MANET routing protocols be proven? Security is an intractable problem, the techniques we have discussed cannot fully prove protocol security. They can only be used to provide operational insight and to analyze MANET routing security weaknesses and stated protocol properties/goals. Table 2 summarizes the capabilities of the presented approaches.

Table 2. Verification Approach Characteristics

Verification Approach	Unknown Attacks	Property Guarantees	Design Phase	Context Dependent
Walkthrough	yes	no	yes	yes
Simulation	no	no	yes	yes
Test	no	no	no	yes
Beta-testing	yes	no	no	yes
Proof	yes	yes	yes	yes
BAN Logic	yes	yes	yes	yes
Formal Methods	yes	yes	yes	no

While we have shown that we can identify some unknown attacks and determine if a protocol property is met, we cannot claim complete security. This problem is analogous to the software engineering approach to testing, we recall that it is impossible to tell if software is defect free from testing. Quoting Dijkstra [22], "Program testing can be used to show the presence of bugs, but never to show their absence!" This is analogous to using any of the presented techniques to prove security, in that the presence of known vulnerabilities can be easily shown, but it is impossible to determine the absence of vulnerabilities. We can find instances that defend against an attack and instances that are vulnerable to attack, but most attacks or vulnerabilities will be unknown until they occur (i.e. in real-world implementations or beta-testing).

One promising solution is the use of automated model checkers to find vulnerabilities. However, this technique can not prove absolute security due to the limitations in state-space explosion in attempting to analyze the infinite possibilities of security vulnerabilities. This is one area for continued research.

Our belief is that the combination of coordinated discussion, simulation, experimentation/testing, analytical proofs, and formal methods/model checkers will provide the most complete security analysis for MANET routing protocols.

REFERENCES

[1] Security Metrics Guide for Information Technology Systems, 2003. NIST SP 800-55, (July)

[2] Sahinoglu, Mehmet. 2005. "Security Meter: A Practical Decision-Tree Model to Quantify Risk." *IEEE Security and Privacy*, (May-June), pp. 18-24.

[3] Abadi, Martín and Phillip Rogaway. 2000 "Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption)." *IFIP International Conference on Theoretical Computer Science*, Lecture Notes in Computer Science, vol. 1872, pp.3-22. Springer.

[4] Syverson, Paul. 1994. ``A Taxonomy of Replay Attacks." *Proceedings of the Computer Security Foundations Workshop VII,* Franconia NH.

[5] Patrikakis, C., M. Masikos, and O. Zouraraki. 2004. "Distributed Denial of Service Attacks." *The Internet Protocol Journal,* (December), vol. 7, no. 4.

[6] Cavin, D., Y. Sasson, and A. Schiper. 2002. "On the Accuracy of MANET Simulators." *In Proceedings of the Second ACM International Workshop on Principles of Mobile Computing*, pp. 38-43.

[7] Takai M., J. Martin, and R. Bagrodia. 2001. "Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks." *In Proceedings of 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, pp. 87-94.

[8] Melendez, Alex "The Monitor and the Principals" , Florida State University CS Technical Report #TR010701

[9] Haines, Joshua W, et al. 1999. "1999 DARPA Intrusion Detection System Evaluation: Design and Procedures." MIT Lincoln Laboratory Technical Report.

[10] Lippmann, Richard P, et al. 2000. "Evaluating Intrusion Detection Systems: the 1998 DARPA Off-Line Intrusion Detection Evaluation." *Proceedings of the 2000 DARPA Information Survivability Conference and Exposition*, Vol. 2, pp 12 – 26.

[11] Awerbuch, Baruch, et al. 2005. "On the Survivability of Routing Protocols in Ad Hoc Wireless Networks." In *Proceedings of the First IEEE Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm 2005)*. Athens (Greece).

[12] Fall, Kevin and Kannan Varadhan, eds. "The NS manual: formerly known as notes and documentation." http://www.isi.edu/nsnam/ns/doc /ns_doc.pdf. Page accessed on Nov 13, 2005.

[13] Kurkowski, S, T. Camp, and M. Colagrosso. 2004. "A Visualization and Animation Tool for NS-2 Wireless Simulations: iNSpect." *Technical Report MCS-04-03, (June)*. The Colorado School of Mines.

[14] Burmester, M., T. Van Le, and A. Yasinsac. 2004. "Adaptive Gossip protocols: Managing Security and Redundancy in Ad Hoc Networks." *3rd International Conference on AD-HOC Networks & Wireless*, LNCS 3158, Springer, July: pp. 96-107.

[15] Burrows, M., M. Abadi, and R. M. Needham. 1990. "A logic of authentication." *ACM Transactions on Computer Systems*, (February), vol. 8, no. 1, pp. 18-36.

[16] Papadimitratos, P. and Z. Haas. 2002. "Secure Routing for Mobile Ad hoc Networks." *In Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, (January 27-31), San Antonio, TX.

[17] Marshall, J., V. Thakur, and A. Yasinsac. 2003. "Identifying flaws in the secure routing protocol." *In Proceedings of the 2003 IEEE International Performance, Computing, and Communications Conference*, (April 9 – 11), .pp. 167- 174.

[18] Thayer Fabrega, F. Javier, Jonathan C. Herzog, and Joshua D. Guttman. 1999. "Strand Spaces: Proving Security Protocols. Correct." *Journal of Computer Security*, vol. 7, pp. 191-230.

[19] Meadows, Catherine A. 1996. "The NRL Protocol Analyzer: An Overview." *Journal of Logic Programming*, (February), vol. 26, no. 2, pp. 133-146.

[20] Yasinsac A. and Wm. A. Wulf. 2001. "A Framework for A Cryptographic Protocol Evaluation Workbench." T*he International Journal of Reliability, Quality and Safety Engineering (IJRQSE)*, (December). vol. 8, no. 4, pp. 373-89.

[21] Holzmann, Gerard J. 1997. "The Model Checker SPIN." *IEEE Transactions on Software Engineering*, (May), vol. 23, no. 5.

[22] Dijkstra, E. W. 1972. "Notes on Structured Programming." *Structured Programming, O.J. Dahl, E.W. Dijkstra, C.A.R.Hoare.* Academic Press, London.

Todd Andel *is a PhD student in the Department of Computer Science at Florida State University. His research interests include wireless security protocols, simulation, and formal methods. He received a master's of science in computer engineering from the Air Force Institute of Technology.*

Dr. Alec Yasinsac *is an Associate Professor of Computer Science at Florida State University. He has worked for twenty six years in the computer science field developing application software, managing and installing systems software on mainframe systems, planning and managing network support for large organizations, and engineering tactical and permanent data networks. He received his doctoral degree from the University of Virginia. He is a Senior Member of IEEE, and a member of the IEEE Computer Society, and the Association of Computing Machines.*

JAN 17 2006

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	13.Jan.06	MAJOR REPORT

4. TITLE AND SUBTITLE
WIRELESS PROTOCOL SECURITY: TO SIMULATE OR NOT TO SIMULATE.

5. FUNDING NUMBERS

6. AUTHOR(S)
CAPT ANDEL TODD R

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
FLORIDA STATE UNIVERSITY

8. PERFORMING ORGANIZATION REPORT NUMBER
CI04-1724

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
THE DEPARTMENT OF THE AIR FORCE
AFIT/CIA, BLDG 125
2950 P STREET
WPAFB OH 45433

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Unlimited distribution
In Accordance With AFI 35-205/AFIT Sup 1

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

14. SUBJECT TERMS	15. NUMBER OF PAGES
	7
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT